

高等院校计算机应用系列教材

数据库技术及应用

(MySQL)

杨宏霞 张婷曼 主 编
段 竹 汪德忠 副主编

清华大学出版社

北 京

内 容 简 介

本书从数据库技术的基本概念入手,通过前后衔接、丰富实用的数据库项目,全面介绍了 MySQL 数据库技术,使读者能够深入浅出、全面系统地掌握 MySQL 数据库管理系统及其应用开发的相关知识。全书以“教务管理系统”案例贯穿始终,从数据库技术的基本理论、数据库设计与实现方法的具体问题开始,讲解数据库和表的创建与管理、视图管理、存储函数、存储过程、触发器、数据库安全管理等内容。此外,本书还基于“学生信息管理系统”详细介绍了使用 Visual Studio 2022 操作 MySQL 数据库的方法,并设计了 16 个课后实验项目。

本书既可作为应用型本科相关专业“数据库技术及应用”课程的配套教材,也可供参加数据库类考试的人员、数据库应用系统开发人员、工程技术人员及其他相关人员参考。对于非计算机专业的本科学生,如果希望学习关键且实用的数据库技术,也可采用本书作为教材。

本书配套的电子课件、习题答案、实例源文件和实验报告模板可以到 <http://www.tupwk.com.cn/downpage> 网站下载,也可以扫描前言中的二维码获取。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。举报:010-62782989, beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

数据库技术及应用:MySQL/杨宏霞,张婷曼主编.

北京:清华大学出版社,2025.2.--(高等院校计算机应用系列教材).--ISBN 978-7-302-67923-3

I. TP311.138

中国国家版本馆 CIP 数据核字第 2025TJ6837 号

责任编辑:胡辰浩

封面设计:高娟妮

版式设计:恒复文化

责任校对:成凤进

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <https://www.tup.com.cn>, <https://www.xuetang.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-83470000

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:河北鹏润印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.25

字 数:416 千字

版 次:2025 年 3 月第 1 版

印 次:2025 年 3 月第 1 次印刷

定 价:69.00 元

产品编号:105280-01

前言

本书在编写过程中，结合党的二十大精神进教材、进课堂、进头脑的要求，将知识教育与思想政治教育有机融合。通过案例加深学生对知识的认识与理解，注重培养学生的创新精神、实践能力和社会责任感。在讲解知识点时，本书将理论知识应用到教学实践中，以动手实践的方式加深学生对知识点的认识与理解。案例设计从现实生活出发，有效激发学生的学习兴趣 and 动手能力，充分发挥学生的主动性和积极性，增强学习信心和学习欲望。同时，在知识讲解中融入素质教育相关内容，引导学生树立正确的世界观、人生观和价值观，进一步提升学生的职业素养，落实德才兼备的高素质卓越工程师和高技能人才的培养要求。

数据库技术是计算机科学的重要领域之一，是计算机数据处理与信息管理的核心技术。随着互联网的迅猛发展，以及云计算、大数据、物联网、区块链、人工智能等新技术的兴起，数据库技术在信息社会中扮演着至关重要的角色，它不仅是有效地组织和存储数据的基础，而且在高效检索大量数据和保障数据安全方面也发挥着关键作用。

数据库技术是信息系统的核心技术之一，它是一种计算机辅助数据管理方法。该技术研究数据的组织和存储方式，以及如何高效地获取和处理数据。目前，数据库技术在各个领域得到了迅速普及和广泛应用。从小型单项事务处理系统到大型信息系统，从联机事务处理到联机分析处理，数据库技术在企业管理、计算机辅助设计与制造、计算机集成制造系统、电子政务、电子商务、地理信息系统等众多领域中发挥了重要的作用。无论是日常的业务管理还是复杂的系统集成，数据库都被用来存储和处理信息资源。

本书汇聚了编者多年从事数据库技术课程教学与应用开发的丰富经验，从数据库技术的基本概念入手，通过前后衔接、丰富实用的数据库项目，全面介绍了 MySQL 8 数据库技术，可以帮助读者深入浅出地掌握 MySQL 数据库管理系统及其应用开发的相关知识。本书内容采用项目驱动的教学方法，使读者真实地感受到现实工作的实际需求，激发学习主动性。通过本书读者不仅能熟练掌握数据库应用的基本知识和技术，还能提升分析问题和解决问题的能力，以及自主学习和掌握计算机新知识、新技术的能力。

本书围绕“教务管理系统”的实施与管理，通过理论与实际相结合的方式，系统讲解了数据库技术的基本理论、数据库设计与实现方法等内容。书中首先介绍了数据库和表的创建与管理、视图管理、数据库安全管理等基本知识，并在解决问题的过程中讲解了数据查询、SQL 语言基础、存储过程与触发器、数据库并发控制等操作技能。此外，本书还基于“学生信息管理系统”详细介绍了使用 Visual Studio 2022 操作 MySQL 数据库的方法，并设计了 16 个课后实验项目，以帮助读者深入理解和应用所学知识。

本书由西安交通大学城市学院多位从事一线教学的教师共同编写完成，其中杨宏霞编写了

第1~3、10章, 段竹编写了第4~5章, 张婷曼编写了6~7章, 汪德忠编写了第8~9章。

鉴于作者水平有限, 书中难免有不足之处, 欢迎广大读者批评指正。在编写本书的过程中参考了相关文献, 在此向这些文献的作者深表感谢。我们的电话是010-62796045, 邮箱是992116@qq.com。

本书配套的电子课件、习题答案、实例源文件和实验报告模板可以到<http://www.tupwk.com.cn/downpage> 网站下载, 也可以扫描下方二维码获取。



编者

2024年8月

目 录

第 1 章 数据库技术基础.....	1	第 3 章 数据库的创建与设计.....	47
1.1 数据库技术概论.....	1	3.1 MySQL数据库管理系统简介.....	47
1.1.1 数据库技术基本概念.....	1	3.1.1 数据库构成.....	47
1.1.2 计算机管理数据技术的发展.....	5	3.1.2 数据库对象.....	49
1.1.3 数据库系统的特点.....	6	3.1.3 数据库对象的标识符.....	49
1.1.4 数据库管理系统的功能.....	6	3.2 SQL语言.....	50
1.2 关系数据库基础.....	7	3.3 管理数据库.....	52
1.2.1 关系模型.....	7	3.3.1 创建数据库.....	52
1.2.2 关系运算.....	10	3.3.2 查看数据库.....	53
1.2.3 关系完整性.....	19	3.3.3 打开或切换数据库.....	54
1.3 数据库发展方向.....	21	3.3.4 修改数据库.....	54
1.4 非关系型数据库.....	22	3.3.5 删除数据库.....	54
1.4.1 非关系型数据库的分类.....	22	3.4 设计数据库.....	55
1.4.2 非关系型数据库的比较.....	23	3.4.1 数据库设计步骤.....	55
1.5 本章小结.....	24	3.4.2 需求分析.....	56
1.6 本章习题.....	24	3.4.3 概念结构设计.....	56
第 2 章 MySQL 概述.....	28	3.4.4 逻辑结构设计.....	59
2.1 MySQL简介.....	28	3.4.5 物理结构设计.....	65
2.2 下载MySQL软件.....	30	3.4.6 数据库实施.....	66
2.3 在Windows中安装MySQL.....	32	3.4.7 数据库运行与维护.....	67
2.4 配置MySQL.....	35	3.4.8 使用MySQL Workbench设计	
2.5 测试MySQL是否安装成功.....	38	数据库.....	68
2.6 MySQL管理工具.....	40	3.5 本章小结.....	71
2.6.1 常用图形化管理工具介绍.....	40	3.6 本章习题.....	71
2.6.2 使用MySQL Workbench管理		第 4 章 表的创建与管理.....	73
数据库.....	41	4.1 表概述.....	73
2.7 本章小结.....	45	4.1.1 表的命名规则.....	73
2.8 本章习题.....	45	4.1.2 常用数据类型.....	74
		4.2 创建和管理表.....	76

4.2.1 表的设计原则和建表步骤	76	6.1.3 日期时间函数	140
4.2.2 创建数据表	78	6.1.4 系统信息函数	144
4.2.3 查看数据表信息	79	6.1.5 自定义函数	146
4.2.4 修改数据表	80	6.2 变量	148
4.2.5 删除数据表	82	6.2.1 变量定义	148
4.3 创建和管理索引	82	6.2.2 变量赋值	149
4.3.1 索引概述	82	6.2.3 系统变量	150
4.3.2 索引的定义与管理	85	6.2.4 会话变量	151
4.3.3 查看索引	88	6.2.5 局部变量	152
4.3.4 删除索引	88	6.3 流程控制语句	153
4.4 关系完整性的实现	89	6.3.1 判断语句	153
4.5 表数据操作	92	6.3.2 循环语句	157
4.6 本章小结	96	6.4 本章小结	159
4.7 本章习题	97	6.5 本章习题	160
第5章 数据查询与视图管理	99	第7章 存储过程和触发器	162
5.1 SELECT语句	99	7.1 存储过程	162
5.2 简单查询	100	7.1.1 创建存储过程	162
5.3 使用聚合函数查询	111	7.1.2 调用存储过程	164
5.4 连接查询	114	7.1.3 查看存储过程	164
5.4.1 内连接	114	7.1.4 修改存储过程	167
5.4.2 自然连接	115	7.1.5 删除存储过程	168
5.4.3 外连接	117	7.2 游标	169
5.4.4 自连接	120	7.2.1 游标操作	169
5.5 子查询	121	7.2.2 游标使用	170
5.5.1 带有ANY或者SOME关键字的子查询	121	7.3 触发器	172
5.5.2 带有ALL关键字的子查询	123	7.3.1 创建触发器	172
5.5.3 带有IN关键字的子查询	124	7.3.2 查看触发器	174
5.5.4 带有比较运算符的子查询	125	7.3.3 删除触发器	175
5.5.5 带有EXISTS关键字的子查询	125	7.4 事件	175
5.6 联合查询	126	7.4.1 开启事件调度器	175
5.7 视图管理	127	7.4.2 创建事件	176
5.8 本章小结	130	7.4.3 查看事件	178
5.9 本章习题	130	7.4.4 修改事件	178
第6章 MySQL 编程基础	133	7.4.5 删除事件	179
6.1 函数	133	7.5 本章小结	179
6.1.1 数学函数	133	7.6 本章习题	180
6.1.2 字符串函数	136	第8章 数据库安全管理	181
		8.1 MySQL的安全性	181

8.1.1 MySQL访问控制工作过程·····	181	10.2 系统设计·····	224
8.1.2 MySQL权限表·····	182	10.2.1 系统功能设计·····	224
8.2 MySQL用户管理·····	183	10.2.2 数据库设计·····	225
8.2.1 创建用户·····	183	10.3 系统实现·····	226
8.2.2 删除用户·····	184	10.3.1 载入数据·····	226
8.2.3 修改用户密码·····	184	10.3.2 数据库接口·····	226
8.3 MySQL权限管理·····	185	10.3.3 搭建开发环境·····	227
8.3.1 授予权限·····	185	10.3.4 添加对MySQL Connector的 引用·····	230
8.3.2 撤销权限·····	189	10.3.5 登录窗体·····	232
8.4 MySQL日志管理·····	190	10.3.6 主窗体·····	234
8.4.1 MySQL日志·····	191	10.3.7 专业信息管理窗体·····	234
8.4.2 二进制日志·····	191	10.4 本章小结·····	235
8.4.3 通用查询日志·····	195	10.5 本章习题·····	236
8.4.4 慢查询日志·····	198	参考文献·····	237
8.5 MySQL数据备份与恢复·····	203	附录 A 实验·····	238
8.5.1 备份数据·····	203	实验1 概念模型设计(绘制E-R图)·····	238
8.5.2 恢复数据·····	207	实验2 逻辑模型设计与完整性·····	239
8.5.3 使用Workbench备份与恢复数据·····	209	实验3 数据库的创建与管理·····	240
8.6 本章小结·····	211	实验4 数据表的创建与管理·····	240
8.7 本章习题·····	211	实验5 数据表约束的管理·····	242
第9章 事务与锁·····	213	实验6 数据插入、修改与删除·····	243
9.1 事务·····	213	实验7 单表数据查询·····	245
9.1.1 事务特性·····	213	实验8 多表数据查询·····	245
9.1.2 事务控制语句·····	214	实验9 视图的创建与管理·····	246
9.2 事务的并发处理·····	217	实验10 MySQL函数应用·····	247
9.3 锁·····	219	实验11 存储过程和游标的使用·····	247
9.3.1 锁机制·····	219	实验12 触发器和事件的使用·····	248
9.3.2 锁的级别·····	219	实验13 数据库的安全管理·····	248
9.3.3 死锁·····	221	实验14 数据的备份与恢复·····	249
9.4 本章小结·····	222	实验15 日志管理·····	249
9.5 本章习题·····	222	实验16 数据库设计·····	250
第10章 综合实例——使用 Visual Studio 2022 操作 MySQL 数据库·····	224		
10.1 需求说明·····	224		

数据库技术基础

数据库技术是一种对计算机数据进行处理和存储的技术，广泛应用于金融、政府、教育、医疗、交通、能源等多个领域。例如，高考志愿填报系统、电子商务平台、教务管理系统、火车票订票系统等场景中都涉及大量的信息存储和不同需求的数据统计与查询。例如，各大电商平台不仅能查询到各种商品的信息，还能实现网上交易，并对消费者的购买行为进行大数据分析，从而向消费者推荐可能感兴趣的物品。

各专业学生都应学习大型数据库的相关知识和技能，以适应未来在数据处理领域的工作要求。本章将介绍数据库的基础知识，包括数据库系统的构成与特点、关系型数据库的基本概念以及关系运算等内容。

1.1 数据库技术概论

本节将介绍数据库中的一些基本概念，包括数据、信息、数据处理、数据库及数据库系统等内容。

1.1.1 数据库技术基本概念

数据、数据库、数据库管理系统和数据库系统是与数据库技术密切相关的基本概念。在学习数据库之前，必须对这些概念有一个深入的理解。

1. 数据

描述事物的符号记录被称为数据。这些符号可以是数字、文字、图形、图像、音频、视频等。数据可以有多种表现形式，它们都可以经过数字化处理后存储在计算机中。

数据的表现形式本身并不足以完全传达其含义，它需要结合相应的解释才能被正确理解。数据和关于数据的解释是不可分的。以数字 88 为例，它可能代表一个学生某门课程中的成绩，也可能表示某个人的体重，或者是某个班的学生总数。

在计算机中，为了存储和处理现实世界中的事物，需要抽象出对这些事物感兴趣的特征，然后将这些特征组成一个记录来进行描述。例如，在学生档案中，如果人们最感兴趣的是学生的姓名、性别、年龄、出生年月、籍贯、所在系别和入学时间，那么可以这样描述：

(张一飞，男，19，2005.08，陕西汉中，计算机系，2024)

这里的学生记录就是数据。对于上面这条学生记录，了解其含义的人可以得到如下信息：张一飞是一名大学生，2005年8月出生，男，籍贯是陕西汉中，2024年考入计算机系。而不了解其语义的人则可能无法准确理解其含义。因此，数据库中存储的数据记录本身无法完全表达其内容，需要经过解释。数据和关于数据的解释是不可分的，数据的解释是指对数据含义的说明，数据的含义称为数据的语义。数据与其语义是密不可分的。

2. 信息

信息是指音讯、消息以及通信系统传输和处理的对象，泛指人类传播的一切内容。人们通过获得和识别自然界及社会的不同信息来区别不同事物，从而得以认识和改造世界。在一切通信和控制系统中，信息是一种普遍联系的形式。1948年，数学家香农在题为《通信的数学理论》的论文中指出：“信息是用来消除随机不定性的东西”。信息被认为是构成宇宙万物的最基本单位。

信息是数据的内涵，它是对数据进行有意义解释的过程。信息是加载于数据之上的，它依赖于数据来表达；而数据则是信息的载体。数据可以是符号、文字、数字、语音、图像、视频等形式，通过这些形式，数据能够生动具体地表达出信息的内容。

简而言之，信息是经过加工处理的数据。它会对接收者的行为和决策产生影响，能够增加决策者的知识，并具有现实的或潜在的价值。

3. 数据处理

数据处理是对数据的采集、存储、检索、加工、变换和传输的过程。

数据处理的基本目的是从大量杂乱无章、难以理解的数据中提取和推导出对于特定人群有价值和有意义的信息。

在处理数据时，目的是将数据转化为信息，为决策和管理提供有价值的依据。而在处理信息时，需要将其转化为数据形式，以便进行存储和传输。

数据处理系统已广泛地用于各种企业和事业单位，内容涉及薪金支付、票据处理、信贷和库存管理、生产调度、计划管理、销售分析等。它能够生成操作报告、财务分析报告和统计报告等。数据处理技术涵盖了文档管理系统、数据库管理系统、分布式数据处理系统等领域的技术。

4. 数据库

数据库(Database, DB)，顾名思义，是存放数据的仓库。这个“仓库”实际上是指计算机存储设备上的数据集合，数据在其中按特定的格式和结构进行组织和存储。

严格地讲，数据库是长期存储在计算机内、有组织的、可共享的大量数据的集合。数据库中的数据按照特定的数据模型进行组织、描述和存储，具有较低的冗余度、较高的数据独立性和良好的扩展性，并可以为各种用户提供共享服务。

在一个数据库中，不仅能保存一种简单的数据，还可以保存不同类型的数据。例如，在进销存管理系统中，可以同时将产品数据和供应商数据保存在同一个数据库文件中，这样在归类及管理时更加方便。若不同数据之间存在关联关系，还可以通过数据库管理系统为不同数据表建立相应关系。例如，可以查询出某种产品的名称、规格和价格，并且可以利用产品的厂商编号查询到厂商名称和联系电话。通常称保存在数据库中的不同类别的记录集合为数据表(Table)，在一个数据库中可以有多个数据表，而数据表之间通常存在各种关联关系，数据库管理系统可

以通过建立数据表之间的关联关系来实现协同作业和相互配合。数据库中数据表之间存在的关联关系如图 1-1 所示。

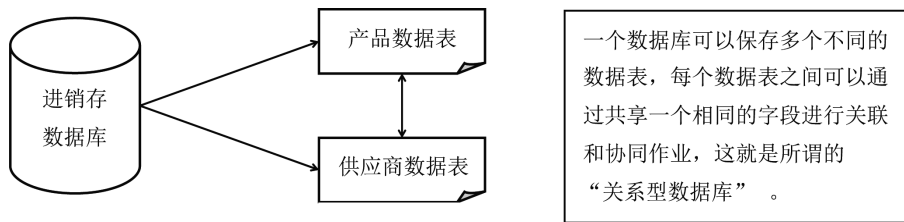


图 1-1 数据表之间关联关系示意图

每一个数据表都是由多个字段组成的。例如，在产品数据表中，可能会包含产品编码、产品名称、价格和供应商编码等字段，只要按照这些字段设置并输入数据，就可以完成一个完整的数据表，如图 1-2 所示。

产品数据表	产品编码	产品名称	价格	供应商编码
	010001	打印机	3500	001
	010002	显示器	2500	001

图 1-2 产品数据表

这里有一个很重要的概念：虽然一般人认为数据库是保存数据的地方，但实际上，数据表才是数据的具体存储单位。数据库是一个容器，用于存放一个或多个数据表。

在关系型数据库中，数据表(Table)是一系列二维数组的集合，用于存储数据和操作数据的逻辑结构。数据表由纵向的列和横向的行组成，行称为记录，是组织数据的单位；列称为字段，每一列表示记录的一个属性，每个字段都有相应的描述信息，例如数据类型、数据宽度等。

5. 数据库系统

数据库系统(Database System, DBS)由数据库、数据库管理系统(及其应用开发工具)、应用系统和数据库管理员(Database Administrator, DBA)组成，负责数据的存储、管理、处理和维护。需要指出的是，数据库的建立、使用和维护等工作仅依靠数据库管理系统是不够的，还需要有专门的人员来完成这些任务，这些人员被称为数据库管理员。

在数据库系统中，数据库负责提供数据的存储功能；数据库管理系统提供数据的组织、存取、管理和维护等基础功能；应用系统根据应用需求使用数据库；而数据库管理员负责全面管理数据库系统。

数据库系统和数据库是两个不同的概念。数据库系统是一个人机系统，数据库是数据库系统中的一个组成部分。然而，在日常工作中，人们常常将“数据库系统”简称为“数据库”。因此，希望读者能够根据上下文区分“数据库系统”和“数据库”，以避免混淆。需要注意的是，数据库管理系统是数据库系统的核心组成部分，如图 1-3 所示。

数据库系统一般由以下 4 个部分组成。

(1) 数据库：是长期存储在计算机内的、具有组织结构，可共享的数据集合。数据库中的数据按照一定的模型进行组织、描述和存储，具有较低的数据冗余、较高的数据独立性和易扩展性，并可为各种用户提供共享服务。

(2) 硬件：构成计算机系统的各种物理设备，包括存储所需的外部设备。硬件的配置应满

足整个数据库系统的需求。

(3) 软件：包括操作系统、数据库管理系统和应用系统。

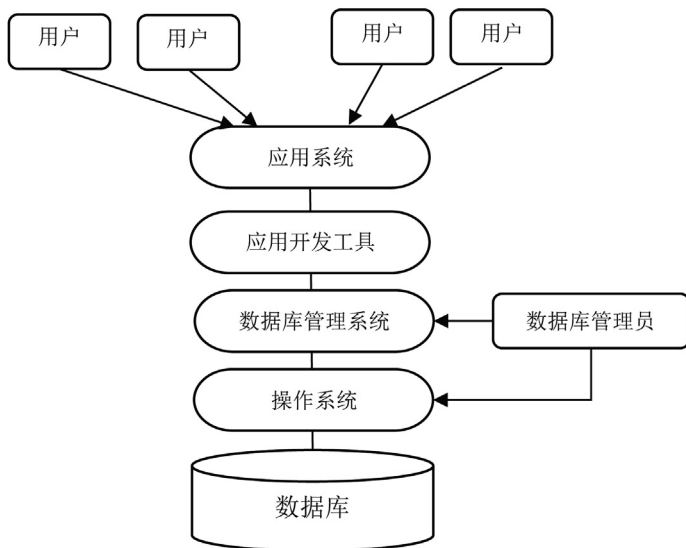


图 1-3 数据库系统

(4) 人员：数据库系统中的人员主要有四类。第一类为系统分析员和数据库设计人员，负责应用系统的需求分析和规范说明，确定需要存放在数据库中的数据，设计数据库的各级模式。第二类为应用程序员，负责编写使用数据库的应用程序。第三类为最终用户，这些用户利用系统的接口或查询语言访问数据库。第四类是数据库管理员，负责数据库的总体管理和信息控制。

6. 数据库管理系统

数据库管理系统(Database Management System, DBMS)是用户创建、管理和维护数据库时所使用的软件，它位于用户与操作系统之间，负责对数据库进行统一管理。DBMS 能够定义数据存储结构，提供数据的操作机制，并维护数据库的安全性、完整性和可靠性。

数据库管理系统类似于图书管理员，通过管理数据库中的数据来实现各种数据管理功能。

7. 数据库应用系统

数据库应用系统(Database Application System, DBAS)不仅为用户提供了数据管理功能，还根据具体应用领域业务规则，通过数据库应用程序，实现了更为复杂的数据处理功能。例如，以数据库为基础的财务管理系统、人事管理系统、图书管理系统等。数据库应用系统负责与 DBMS 进行通信，访问和管理 DBMS 中存储的数据，并允许用户插入、修改和删除数据库中的数据。

综上所述，数据库(DB)、数据库管理系统(DBMS)、数据库应用系统(DBAS)和数据库系统(DBS)是四个不同的概念。数据库系统(DBS)是一个整体系统，包含数据库(DB)及数据库管理系统(DBMS)和数据库应用系统(DBAS)。数据库管理系统是在操作系统支持下对数据库进行管理的工具软件。数据库管理系统如同一座桥梁，一端连接面向用户的数据库应用系统，另一端连

接存放数据的数据库。数据库管理系统是数据库系统的核心。

1.1.2 计算机管理数据技术的发展

随着计算机软件和硬件的发展,计算机管理数据技术经历了人工管理、文件系统管理和数据库系统管理三个阶段。每一阶段的发展以数据存储冗余不断减小、数据独立性不断增强、数据操作更加方便和简单为标志,各有各的特点。

1. 人工管理阶段

20 世纪 50 年代中期以前,计算机主要用于科学计算。硬件方面没有磁盘来存储数据,因此数据和程序必须结合在一起,数据不独立。在计算某一课题时,程序和对应的数据一起装入计算机,计算完毕后数据不会保存。软件方面没有专门管理数据的软件,数据管理只能依赖人工(程序员)进行,因此称为人工管理阶段。

在人工管理数据阶段,程序员将程序和数据编写在一起,每个程序都有属于自己的一组数据,数据与应用程序之间是一一对应的。因此,数据不能共享,即便是几个程序处理同一批数据,运行时也必须重复输入,导致程序间存在大量的重复数据,称为数据冗余。另外,数据的存储结构、存取方法、输入输出方式完全由程序员在程序中自行设计和安排,改变数据时必然要修改程序。

2. 文件系统管理阶段

20 世纪 50 年代后期到 60 年代中期,计算机开始大量应用于管理方面。此时,硬件方面的磁盘成为主要的外存设备,这样大量需要长期保留的数据就可以与程序分离,并以数据文件的形式独立保存在计算机的外存上,以便对其进行反复处理。在操作系统中,也出现了专门的数据管理软件,称为文件系统。用户只需要知道数据文件的名称,而不必了解数据存放的位置以及如何存储。文件系统就可以将相应的数据提供给用户使用,实现了“按文件名进行访问、按记录进行存取”的管理技术。

在文件系统管理数据阶段,数据相对于程序具有一定的独立性,可以以文件的形式独立保存。不同的程序可以处理存储在硬盘上的同一个数据文件,程序与数据之间形成了多对多的关系。一组数据可以被多个程序使用,一个程序也可以使用多组数据,实现了数据共享。

但是在文件系统中,数据共享只能以文件为单位,这导致不同的数据文件中可能会出现大量的重复数据,这不仅浪费了存储空间,同时也给数据的维护带来困难。为解决数据的独立性问题,实现数据的统一管理,达到数据共享的目的,数据库技术应运而生并不断发展。

3. 数据库系统管理阶段

在 20 世纪 60 年代后期以来,计算机性能得到很大提高,应用也越来越广泛。同时,硬件方面出现了大容量且价格低廉的磁盘,软件方面操作系统也开始成熟。在这种背景下,以文件系统作为数据管理手段已经不能满足应用的需求。为了解决多用户、多应用共享数据的需求,使数据为尽可能多地应用服务,数据库技术便应运而生,出现了统一管理数据的专门软件系统——数据库管理系统。这将传统的数据管理技术推向一个新阶段,即数据库系统管理阶段。

在这一阶段,数据库系统对数据的处理方式是将所有应用程序中使用的数据汇集在一起,使数据的组织和管理与具体的应用相脱离,全部交由数据库管理系统统一管理。这样不仅实现

了数据与程序的完全独立，而且大大减少了数据冗余，真正实现了数据的共享。

1.1.3 数据库系统的特点

数据库系统的特点包括：数据结构化、数据共享性高、冗余度低、易于扩展、数据独立性高，以及数据由 DBMS 统一管理和控制。

1) 数据结构化

数据库系统实现了整体数据的结构化，这是数据库的最主要的特征之一。这里所说的“整体”结构化，是指在数据库中的数据不再仅针对某个特定应用，而是面向整个组织；不仅数据内部是结构化的，而且整个数据集也是结构化的，数据之间存在联系。

数据结构化是数据库系统与文件系统的根本区别。在文件系统中，独立文件内部的数据一般是有结构的，但文件之间通常不存在联系，因此从整体上看，数据是没有结构的。

2) 数据共享性强，冗余度低，易于扩展

因为数据是面向整体的，所以可以被多个用户和多个应用程序共享使用，这样可以大大减少数据冗余，节约存储空间，避免数据之间的不相容性和不一致性。

3) 数据独立性高

数据的独立性是数据库系统的最基本的特征之一。数据独立性是指应用程序与数据结构之间相互独立，互不影响。数据独立性包括数据的物理独立性和逻辑独立性。

物理独立性是指数据在磁盘上的存储方法由数据库管理系统 DBMS 管理，应用程序无需了解。应用程序只需处理逻辑结构，因此当数据的物理存储结构发生改变时，应用程序无需改变。

逻辑独立性是指应用程序与数据库的逻辑结构相互独立。也就是说，即使数据的逻辑结构发生了改变，应用程序也可以不进行修改。

4) 数据由 DBMS 统一管理和控制

数据库管理系统(DBMS)用于建立、使用和维护数据库。它对数据库进行统一的管理和控制，以保证数据库的安全性和完整性。DBMS 使多个应用程序和用户可以采用不同的方法同时或在不同时间建立、修改和访问数据库。

1.1.4 数据库管理系统的功能

数据库的共享是并发的共享，即多个用户可以同时访问数据库中的数据，甚至可以同时访问数据库中的同一个数据。DBMS 必须提供以下几方面的数据控制功能。

(1) 数据定义：DBMS 提供数据定义语言(Data Definition Language, DDL)，主要用于定义数据库的结构、数据之间的关系等。

(2) 数据操作：DBMS 提供数据操作语言(Data Manipulation Language, DML)，实现对数据的插入、删除、更新、查询等操作。

(3) 数据库的运行管理：数据库的运行管理功能是 DBMS 的运行控制和管理功能，包括多用户环境下的并发控制、安全性检查和存取限制控制、完整性检查和执行、运行日志的组织和管理、事务的管理和自动恢复(即保证事务的原子性)。这些功能保证了数据库系统的正常运行。

(4) 数据组织、存储与管理：DBMS 要分类组织、存储和管理各种数据，包括数据字典、用户数据和存取路径等。需确定以何种文件结构和存取方式在存储级上组织这些数据，以及如

何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率，选择合适的存取方法提高存取效率。

(5) 数据库的保护：数据库中的数据是信息社会的战略资源，因此数据的保护至关重要。DBMS 通过以下四个方面实现对数据库的保护：数据库恢复、数据库并发控制、数据完整性控制和数据库安全性控制。DBMS 的其他保护功能还包括系统缓冲区的管理以及数据存储的一些自适应调节机制等。

(6) 数据库的维护：这一部分包括数据库的数据载入、转换、转储、数据库的重组和重构以及性能监控等功能，这些功能通常不同的应用程序或工具来完成。

(7) 通信：DBMS 具有与操作系统的联机处理、分时系统及远程作业输入的相关接口，负责处理数据的传送。对网络环境下的数据库系统，还应该包括 DBMS 与网络中其他软件系统的通信功能以及数据库之间的互操作功能。

1.2 关系数据库基础

MySQL 数据库属于关系型数据库，因此读者需要了解关系型数据模型的相关知识。

数据模型的种类有很多，目前被广泛使用的主要有两种类型。一种是独立于计算机系统的数据模型，它完全不涉及信息在计算机中的表示，仅用来描述某个特定组织所关心的信息结构，这种模型称为“概念数据模型”。概念数据模型是按照用户的观点对数据进行建模，强调其语义表达能力。它应该简单、清晰、易于用户理解，是对现实世界的第一层抽象，也是用户与数据库设计人员之间进行交流的工具，并且是进行数据库设计的重要工具。其典型代表就是著名的“实体关系模型”(E-R 模型)。

另一种数据模型是直接面向数据库的逻辑结构，它是对现实世界的第二层抽象。这种模型直接与数据库管理系统相关，称为“逻辑数据模型”，包括层次模型、网状模型、关系模型和面向对象模型。

在 4 种逻辑数据模型中，层次模型和网状模型已经很少应用，而面向对象模型比较复杂，尚未达到关系模型数据库的普及程度。目前，理论成熟、使用广泛的模型是关系模型。自 20 世纪 80 年代以来，计算机厂商推出的数据库管理系统几乎都是支持关系模型的，关系模型已经占据市场的主导地位。

1.2.1 关系模型

关系模型是由若干个关系模式组成的集合，关系模式的实例称为关系，每个关系实际上是一张二维表格。关系模型通过键来导航数据，其表格结构简单，用户只需用简单的查询语句就可以对数据库进行操作，并不涉及存储结构、访问技术等细节。SQL 语言是关系数据库的代表性语言，已经得到了广泛的应用。典型的关系数据库产品有 MySQL、DB2、Oracle、Sybase、SQL Server 等。

关系型数据库是以关系模型为基础的数据库，根据表、元组、字段之间的关系进行组织和访问数据。它通过若干张表来存取数据，并且通过关系将这些表联系在一起。关系型数据库是目前应用最广泛的数据库类型。关系型数据库是支持关系模型的数据库，下面将介绍关系模型。

关系模型有 3 个组成部分：数据结构、数据操作和完整性规则。关系模型建立在严格的数学概念基础之上，利用二维表来描述实体及实体之间的联系。

例如，在一个名为 `student` 的学生信息表中，每一列都包含了所有学生的某个特定类型的信息，如姓名；而每一行则包含了某个特定学生的所有信息，包括学号、姓名、性别和专业，如图 1-4 所示。

学号	姓名	性别	专业
23020001	张岳阳	男	计算机科学与技术
23020002	李芳	女	网路工程
23020003	王明	男	软件工程

图 1-4 学生信息表(student)

1. 关系模型相关术语

1) 关系

一个关系就是一张由行与列构成的二维表，每张表具有一个表名，即关系名。不同的实体集必须用不同的表来表示，不能保存在同一张表中，即表之间是不允许嵌套的。例如，图 1-4 展示的是一个 3 行 4 列的学生关系(学生表)，对应一个学生实体集。

2) 属性

表中的列称为属性或字段，列的名称称为属性名，在列中填写的数据称为属性值。如图 1-4 所示，学生表有 4 列，分别对应 4 个属性：学号、姓名、性别和专业。

3) 元组

表中第二行开始的每一行称为元组，即通常所说的记录，是构成关系的一个个实体。因此，“关系”是“元组”的集合，而“元组”是属性值的集合。在关系模型中，数据就是这样逐行逐列地组织起来的。如图 1-4 所示，其中的每一行记录都代表客观世界的某个实体，对应于现实世界的一名具体学生。

4) 分量

元组中的每一个属性值称为元组的分量。

5) 域

属性的取值范围称为域，是一组具有相同数据类型的值的集合。例如，学生性别的域是{男，女}，而专业的域则是学校所有专业的集合。

6) 关键字

关系中的某一个属性或某几个属性的组合，如果能够唯一地标识关系中的各个元组且不含多余的属性，则该属性(组)称为该关系的关键字(Key)，也称为“键”，对应概念模型中的“码”。关键字是一个字段或字段的组合。在一张表中可以选定一个关键字作为主关键字(Primary Key)，而其他关键字则称为候选关键字(Candidate Key)。

7) 关系模式

二维表的第一行即表头，确定了一个二维表的结构。例如，表头定义了表的列数，每一列

的名称, 以及每一列能够填写的数据类型等。如图 1-4 所示, 学生表中的学号、姓名、性别等字段及其相应的数据类型组成了学生表的结构。在关系模型中, 表头称为关系模式, 可以使用关系名(属性名 1, 属性名 2, ..., 属性名 n)的形式描述一个关系。例如, 学生关系的关系模式可以表示为:

学生(学号, 姓名, 性别, 专业)

一个关系模型是由若干个关系模式组成的集合。在关系模型中, 实体以及实体间的联系都用关系来表示。例如, 学生、课程以及学生与课程之间的多对多联系可以在关系模型中按如下表示:

学生(学号, 姓名, 性别, 专业)

课程(课程号, 课程名, 学分)

选修(学号, 课程号, 成绩)

由于关系模型概念简单、清晰、易懂、易用, 并且具有严密的数学基础以及在此基础上发展起来的关系数据理论, 简化了程序开发及数据库建立的工作量, 因此迅速获得了广泛的应用, 并在数据库系统中占据了主导地位。

2. 关系的特点

从形式上看, 一个关系可以看作一个二维表, 但并不是所有的二维表都可以称为关系。只有满足以下要求和限制的二维表才可以称为关系。

1) 关系的每一个分量都必须是不可分的数据项

在概念世界中, 实体与属性的区别在于, 属性是实体所具有的某一特征, 一个实体由若干个属性来描述, 而属性不能再分解成更小的特征。这一区别在数据世界中表现为关系中的所有属性值都是不可再分的最小数据单元。在如图 1-5 所示的表格中, 入库、出库、结余又分别有数量、单价和金额三个分项, 因此这是一个复合表, 不是关系。

日期	入库			出库			结余		
	数量	单价	金额	数量	单价	金额	数量	单价	金额
2024-03-01	50	1.20	60.00				50	1.20	60.00
2024-03-02	12	1.30	15.60	51	1.20	61.20	62	1.22	75.64
2024-03-03				10	1.30	13.00	11	1.30	14.30
2024-03-04							1	1.30	1.30
2024-03-05	34	1.40	47.60				35	1.39	48.65

图 1-5 复合表

2) 关系中同一列的数据类型必须相同

表中每一列的取值范围应属于同一类型的数据, 来自同一个域。例如, 学生选课成绩表中的成绩属性值不能有有的是百分制、有的是五分制, 而必须统一为一种表示语义(比如都用百分制), 否则可能会出现存储和数据操作错误。

3) 在同一个关系中不允许出现相同的属性名

定义表结构时, 一张表中不能出现重复的字段名。因为这样的重复不仅会导致数据冗余, 还会产生列标识混乱的问题。

4) 在一个关系中, 列的次序无关紧要

交换表中任意两列的位置并不影响数据的实际含义。例如, 在工资表中, 奖金和基本工资哪一列放在前面都不会改变数据的意义。

5) 在一个关系中, 元组的次序无关紧要

二维表的表体部分是由填写在表格中的数据构成的, 一条记录由若干个字段的值组成, 所有的记录形成一张表, 即表是记录的“容器”。数据库中的数据是不断更新的, 比如可以对表中的数据进行添加、删除、修改等操作, 因此表体部分通常是随着表的使用过程而不断变化的。在使用表的过程中, 可以根据各种排序要求对元组的次序进行重新排列, 例如, 可以分别对教师表中的记录按教工编号的值升序或降序重新排序。因此, 在一张表中任意交换两行的位置并不影响数据的实际含义。

6) 在同一个关系中不允许有完全相同的元组

表中的每一行必须是唯一的, 即表中任意两行不能完全相同。

上述6条性质构成了关系的基本定义, 也是用来判断某个表格是否满足关系模型的基本标准。

1.2.2 关系运算

使用数据库的目的就是为了能够随时找到感兴趣的数据。在关系数据库中, 用户只需明确提出“要做什么”, 而不需要指出“怎么去做”, 系统将自动优化查询过程, 并能够实现对多个相关联的表的高效存取。然而, 正确表示复杂的查询并非是一件简单的事。在关系数据库中, 有些查询操作可能需要多个基本运算的组合。了解关系运算有助于更准确地表达查询需求。

关系运算的运算对象和运算结果都是关系, 即元组的集合。根据运算符的不同, 关系运算可分为传统的集合运算和专门的关系运算两类。传统的集合运算包括并、交、差和广义笛卡儿积四种。专门的关系运算主要有选择、投影、连接和除法等。

1. 关系运算符

在两类运算中, 将用到以下两类辅助操作符。

(1) 比较运算符: $>$ 、 $>=$ 、 $<$ 、 $<=$ 、 $=$ 、 $\neq$ 。

(2) 逻辑运算符: $\vee$ (或)、 $\wedge$ (与)、 $\neg$ (非)。

2. 传统的集合运算

传统的集合运算把关系视为元组的集合, 以元组作为集合中的元素进行运算, 这些运算是从关系的“水平”方向即行的角度进行的。传统的集合运算包括并($\cup$)、交($\cap$)、差($-$)和广义笛卡儿积($\times$)四种运算。

1) 并运算

关系 R 和关系 S 的并运算的结果由属于 R 或属于 S 的元组组成的集合。即将 R 和 S 所有元组合并, 再去除重复的元组, 组成一个新的关系称为 R 和 S 的并, 记为 $R \cup S$ 。形式定义如下:

$$R \cup S = \{t | t \in R \vee t \in S\}$$

例如, 关系 R 中的元组为英语 90 分以上的学生, 关系 S 中的元组为数学 90 分以上的学生。这两个关系结构相同, 仅名称不同, 关系中的记录不同。则 $R \cup S$ 结果就是单科 90 分以上的学

生，如图 1-6 所示。

R				S			
学号	姓名	性别	班级	学号	姓名	性别	班级
23050002	薛佳宏	男	会计 01	23020015	王文秀	女	网路 01
23040015	韩金效	男	土木 02	23060025	韩豆豆	女	软件 03
23010026	任一格	男	电信 01	23050002	薛佳宏	男	会计 01

$R \cup S$			
学号	姓名	性别	班级
23050002	薛佳宏	男	会计 01
23040015	韩金效	男	土木 02
23010026	任一格	男	电信 01
23020015	王文秀	女	网路 01
23060025	韩豆豆	女	软件 03

图 1-6 并运算

注意：

两个关系的并集 $R \cup S$ 仍然是一个关系，依然需要遵守关系的性质，即在同一个关系中不允许有完全相同的元组。因此，在合并两个关系的元组时，若有完全相同的元组，只保留一个。

2) 交运算

关系 R 和关系 S 的交运算是由既属于 R 又属于 S 的元组组成的集合。即在两个关系 R 和 S 中取相同的元组，组成一个新关系，称为 R 和 S 的交，记为 $R \cap S$ 。形式定义如下：

$$R \cap S = \{t | t \in R \wedge t \in S\}$$

例如，关系 R 中的元组为英语 90 分以上的学生，关系 S 中的元组代表数学 90 分以上的学生，那么 $R \cap S$ 的结果就是英语和数学都是 90 分以上的学生，如图 1-7 所示。

R				S			
学号	姓名	性别	班级	学号	姓名	性别	班级
23050002	薛佳宏	男	会计 01	23020015	王文秀	女	网路 01
23040015	韩金效	男	土木 02	23060025	韩豆豆	女	软件 03
23010026	任一格	男	电信 01	23050002	薛佳宏	男	会计 01

$R \cap S$			
学号	姓名	性别	班级
23050002	薛佳宏	男	会计 01

图 1-7 交运算

3) 差运算

关系 R 和关系 S 的差运算是由属于 R 而不属于 S 的元组组成的集合。即在关系 R 中删去与 S 关系中相同的元组，组成一个新的关系，记为 $R - S$ 。形式定义如下：

$$R - S = \{t \mid t \in R \wedge \neg t \in S\}$$

例如, 关系 R 中的元组为英语 90 分以上的学生, 关系 S 中的元组为数学 90 分以上的学生, 则 $R-S$ 结果就是英语 90 分以上但数学分数不在 90 分以上的学生, 如图 1-8 所示。

R				S			
学号	姓名	性别	班级	学号	姓名	性别	班级
23050002	薛佳宏	男	会计 01	23020015	王文秀	女	网路 01
23040015	韩金效	男	土木 02	23060025	韩豆豆	女	软件 03
23010026	任一格	男	电信 01	23050002	薛佳宏	男	会计 01

$R-S$			
学号	姓名	性别	班级
23040015	韩金效	男	土木 02
23010026	任一格	男	电信 01

图 1-8 差运算

注意:

在差运算中, 运算对象的顺序不同会导致不同的结果。在这个例子中, $R-S$ 和 $S-R$ 的结果是不相同的。

4) 广义笛卡儿积运算

两个分别具有 n 列和 m 列的关系 R 和 S 的广义笛卡儿积是一个包含 $n+m$ 列的元组集合, 其中每个元组的前 n 列来自关系 R 的一个元组, 后 m 列来自关系 S 的一个元组。若 R 有 k_1 个元组, S 有 k_2 个元组, 则关系 R 和关系 S 的广义笛卡儿积有 $k_1 \times k_2$ 个元组, 记作 $R \times S$ 。形式定义如下。

$$R \times S = \{t_r \frown t_s \mid t_r \in R \wedge t_s \in S\}$$

例如, 关系 R 和 S 分别具有 3 个属性列, $R \times S$ 的结果如图 1-9 所示。

简单来说, 就是把 R 表的第一行与 S 表第一行组合写在一起, 作为一行。然后把 R 表的第一行与 S 表第二行依次写在一起, 作为新一行。以此类推。当 S 表的每一行都与 R 表的第一行组合过一次以后, 换 R 表的第二行与 S 表第一行组合, 以此类推, 直到 R 表与 S 表的每一行都组合过一次, 则运算完毕。

如果 R 表有 n 行, S 表有 m 行, 那么笛卡儿积 $R \times S$ 有 $n \times m$ 行。

提示:

没有限制的表连接就是笛卡儿积。

R			S	
学号	姓名	性别	课程号	课程名
23050002	薛佳宏	男	01	C 语言
23040015	韩金效	男	02	网络工程
23010026	任一格	男	03	软件工程
23020015	王文秀	女		
23060025	韩豆豆	女		

R×S				
学号	姓名	性别	课程号	课程名
23050002	薛佳宏	男	01	C 语言
23050002	薛佳宏	男	02	网络工程
23050002	薛佳宏	男	03	软件工程
23040015	韩金效	男	01	C 语言
23040015	韩金效	男	02	网络工程
23040015	韩金效	男	03	软件工程
23010026	任一格	男	01	C 语言
23010026	任一格	男	02	网络工程
23010026	任一格	男	03	软件工程
23020015	王文秀	女	01	C 语言
23020015	王文秀	女	02	网络工程
23020015	王文秀	女	03	软件工程
23060025	韩豆豆	女	01	C 语言
23060025	韩豆豆	女	02	网络工程
23060025	韩豆豆	女	03	软件工程

图 1-9 广义笛卡儿积运算

3. 专门的关系运算

专门的关系运算不仅包括行运算,还包括列运算,这种运算是为数据库应用而引进的特殊操作,包括选择运算(σ)、投影运算(Π)、连接运算($\bowtie$)和除法运算($\div$)等。假设有一个数据库,其中包括课程关系 C、学生关系 S、选修关系 SC,如图 1-10 所示。下面的例子将对这个三个关系进行运算。

C			S				
课程号 CNo	课程名 Cname	学分 Ccredit	学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
1	数据库技术及应用	3	95001	李勇	男	20	CS
2	高等数学	3	95002	刘晨	女	19	IS
3	软件工程	4	95003	王敏	女	18	MA
			95004	张智	男	19	IS
			95005	刘明	男	19	AR
			95006	王芳	女	19	IS
			95007	郭慧	女	19	IS

SC		
学号 Sno	课程号 Cno	成绩 Grade
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80

图 1-10 数据库示例

1) 选择

在关系 R 中选择满足条件的元组的操作称为选择运算，即在表中选择满足某些条件的行。在关系 R 中选择满足给定条件的诸元组，记作：

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t)\}$$

- F ：选择条件，是一个逻辑表达式，取值为“真”或“假”，基本形式为 $X_1 \theta Y_1$ 。
 - θ ：表示比较运算符，可以是 $>$ ， $>=$ ， $<$ ， $<=$ ， $=$ ， $\neq$ 。
 - X_1 、 Y_1 ：可以是属性名、常量或简单函数，属性名也可以用它的序号来代替。
- 选择运算是从关系 R 中选取使逻辑表达式 F 为真的元组，是从行的角度进行的运算。

【例 1-1】查询信息系(IS 系)全体学生。

$$\sigma_{Sdept = 'IS'}(S) \text{ 或 } \sigma_{5 = 'IS'}(S)$$

查询结果如图 1-11 所示。

查询信息系(IS 系)全体学生				
学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
95002	刘晨	女	19	IS
95004	张智	男	19	IS
95006	王芳	女	19	IS
95007	郭慧	女	19	IS

图 1-11 查询信息系全体学生

【例 1-2】查询年龄小于 20 岁的学生。

$$\sigma_{Sage < 20}(S) \text{ 或 } \sigma_{4 < 20}(S)$$

查询结果如图 1-12 所示。

查询查询年龄小于 20 岁的学生				
学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张智	男	19	IS
95005	刘明	男	19	AR
95006	王芳	女	19	AR
95007	郭慧	女	19	EC

图 1-12 查询小于 20 岁的学生

2) 投影

从关系 R 中选择若干属性得到无重复元组的新的关系，称为投影，记作：

$$\pi_A(R) = \{t[A] \mid t \in R\}$$

其中 A 为需要从关系 R 中选取的属性列表。

【例 1-3】查询学生的姓名和所在系。

$$\pi_{Sname, sdept}(S) \text{ 或 } \pi_{2,5}(S)$$

查询结果如图 1-13 所示。

查询学生的姓名和所在系	
姓名 Sname	所在系 Sdept
李勇	CS
刘晨	IS
王敏	MA
张智	IS
刘明	AR
王芳	IS
郭慧	IS

图 1-13 查询学生姓名和所在系

注意：

投影操作不仅取消了原关系中的某些列，而且可能取消某些元组。因为取消了某些属性列之后，就可能出现重复行(应取消这些完全相同的行)。

3) 连接

连接又称为 θ 连接，可以分为条件连接与自然连接。连接操作是从两个关系的笛卡儿积中选择满足特定条件的元组，以形成一个新的关系。条件连接的表示形式为：

$$R \bowtie_{x\theta y} S = \{t_r \cap t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[X]\theta t_s[Y]\}$$

- θ 连接是二目运算，它是从两个关系的笛卡儿积中选择满足条件的元组，组成新的关系。
- θ 是运算符的代指，它可以是 $>$ 、 $>=$ 、 $<$ 、 $<=$ 、 $=$ 、 $\neq$ 。
- 等值连接是 θ 连接中的一种特殊的连接，当 θ 为 “=” 时的连接运算称为等值连接，它是从关系 R 与 S 的笛卡儿积中选取属性值相等的那些元组。
- θ 连接的本质是广义笛卡儿积运算加选择运算。即先对两个关系 R 和 S 做广义笛卡儿积运算产生新的关系 N ，再在 N 中选取满足条件的元组，并形成一个新的关系。运算过程如图 1-14 所示。

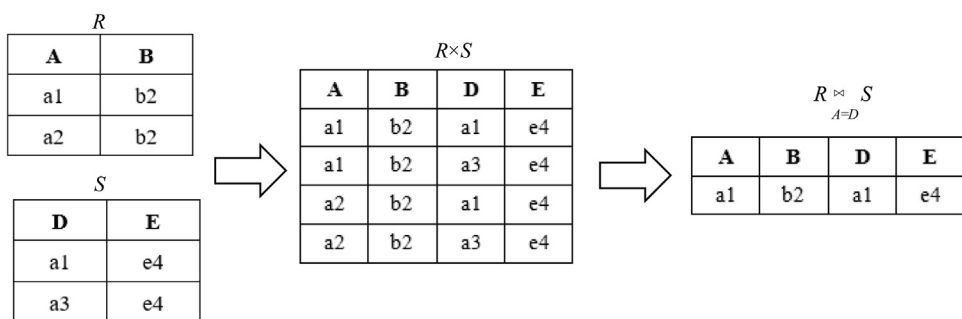


图 1-14 连接运算过程

4) 自然连接

自然连接是等值连接的一种特殊情况，当两个关系中的连接属性具有相同的属性名时，自然连接会去掉连接结果中重复的属性列，形成一个新的表。自然连接的表示形式为：

$$R \bowtie S = \{t_r \cap t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[Y] = t_s[Y]\}$$

- 自然连接的本质是用相同属性进行等值连接，并在结果中去掉重复的属性列。
- 自然连接要求连接的属性名相同，而等值连接不需要属性名相同，只要求属性值取自同一个域。

自然连接就是去掉重复列的等值连接，自然连接运算过程如图 1-15 所示。

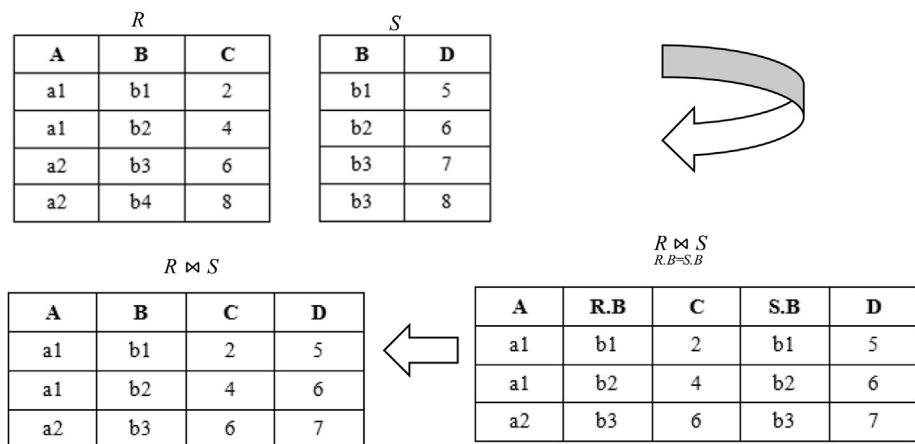


图 1-15 自然连接运算过程

提示：

若两个关系 R 与 S 中有同名属性，为了区分这些属性，通常在属性名前分别加上关系名作为前缀，关系名和属性间用点符号“.”隔开(例如 $R.B$ 或 $S.B$)。

5) 除

除法运算是二目运算，设有关系 $R(X, Y)$ 与关系 $S(Y, Z)$ ，其中 X, Y, Z 为属性集合， R 中的 Y 与 S 中的 Y 可以有不同的属性名，但对应属性值必须出自相同的域。

除运算求解过程如下。

第一步：找出关系 R 和关系 S 中相同的属性，即 Y 属性。在关系 S 中对 Y 做投影(即将 Y 列取出)。

第二步：在关系 R 中，对剩余的属性 X 做投影，并确保这些属性值是唯一的。

第三步：求关系 R 中 X 属性对应的像集 Y 。

第四步：判断包含关系， $R \div S$ 其实就是判断关系 R 中 X 各个值的像集 Y 是否包含关系 S 中属性 Y 的所有值。

设有关系 $R(X, Y)$ 和 $S(Y)$ ， $R \div S$ 运算结果如图 1-16 所示。

R			S		
A	B	C	B	C	D
a1	b1	c2	b1	c2	d1
a2	b3	c7	b2	c1	d1
a3	b4	c6	b2	c3	d2
a1	b2	c3			
a4	b6	c6			
a2	b2	c3			
a1	b2	c1			

$R \div S$	
A	
a1	

图 1-16 除运算

为了计算 $R \div S$ ，需要引进“象集”的概念，具体意义如下所述。

关系 R 和关系 S 拥有共同的属性 B 、 C ， $R \div S$ 得到的就是关系 R 包含而关系 S 不包含的属性，即 A 属性。

在 R 关系中 A 属性的值可以取 $\{a1, a2, a3, a4\}$ 。

- $a1$ 对应的象集为 $\{(b1, c2), (b2, c3), (b2, c1)\}$ 。
- $a2$ 对应的象集为 $\{(b3, c7), (b2, c3)\}$ 。
- $a3$ 对应的象集为 $\{(b4, c6)\}$ 。
- $a4$ 值对应的象集为 $\{(b6, c6)\}$ 。

关系 S 在 B 、 C 上的投影为 $\{(b1, c2), (b2, c1), (b2, c3)\}$ 。

那么，只有 $a1$ 对应的象集包含关系 S 的投影，所以只有 $a1$ 包含在 A 属性中，所以 $R \div S$ 为 $a1$ 。

选择、投影和连接是关系数据语言最基本的运算。也就是说，至少能够提供这三种运算功能的数据库语言才能称为关系数据语言。

4. 关系运算综合练习

已知有如下 4 个关系(表)。

- 教师表: T(TID, TNAME, TITLE)
- 课程表: C(CID, CN, CNAME, TID)
- 学生表: S(SID, SNAME, AGE, SEX)
- 选课表: SC(SID, CID, SCORE)

其中，有下画线的字段是主键。

现在需要用关系代数来实现例 1-4 至例 1-9 的运算。

【例 1-4】 查询课程号为 C2 的学生的学号和成绩。

给定某个条件的查询，就是做选择运算，然后再对选择结果进行一次投影即可(注意，通常优先做选择运算)。

$$\pi_{SID,SCORE}(\sigma_{CID='C2'}(SC))$$

【例 1-5】 查询课程号为 C2 的学生的学号和姓名。

进行一次选择之后，发现 SC 表里面没有姓名这个属性，如果需要姓名属性就需要到表 S 里面去查询，这时就需要到自然连接(JOIN)操作了。

$$\pi_{SID,SNAME}(\sigma_{CID='C2'}(SC \bowtie S))$$

【例 1-6】 检索至少选修了一门 Liu 老师讲授的课程的学生的学号和姓名。

$$\pi_{SID, SNAME}(\sigma_{TNAME='Liu'}(T \bowtie C \bowtie SC \bowtie S))$$

【例 1-7】 查询选修课程号为 C2 或者 C4 的学生的学号。

$$\pi_{SID}(\sigma_{CID='C2'} \vee \sigma_{CID='C4'}(SC))$$

【例 1-8】 易错题：查询选修课程号是 C2 和 C4 课程的学生的学号。
是否可以把 1-7 中的 $\vee$ 换成 $\wedge$ ？如果关系代数表达式写成以下样式：

$$\pi_{SID}(\sigma_{CID='C2'} \wedge \sigma_{CID='C4'}(SC))$$

那么，关系运算表达式将出现歧义。表达式中的 CID 到底应该等于 C2 还是等于 C4？因为 CID 只能等于一个值。如果要选出选修 C2 的学生和选修 C4 的学生，则关系运算表达式应该写成如下样式：

$$\pi_{SID}(\sigma_{CID='C2'}(SC) \cup \sigma_{CID='C4'}(SC))$$

【例 1-9】 查询不学 C2 课程的学生姓名和年龄。
先选出学 C2 课程的学生。

$$\pi_{SNAME,AGE}(\sigma_{CID='C2'}(SC \bowtie S))$$

然后，用全部学生减去学 C2 课程的学生即可。

$$\pi_{SNAME,AGE}(S) - \pi_{SNAME,AGE}(\sigma_{CID='C2'}(SC \bowtie S))$$

1.2.3 关系完整性

数据库中的数据是从外界输入的,输入数据时可能会发生意外,如输入无效或错误信息。在多用户的关系数据库系统中确保输入数据的合规性是重要的考虑因素之一。因此,在设计数据库时,最重要的是确保数据正确存储到数据库的表中。

关系完整性指的是关系数据库中数据的正确性、相容性和有效性。它是给定的关系模型中数据及其联系的所有制约和依存规则,用以限定数据库状态及状态变化,从而保证数据的正确、相容和有效。关系完整性通过各种各样的完整性约束来保证,因此关系完整性设计就是关系完整性约束的设计。现代的数据库管理系统在不同程度上支持完整性规则检查,关系完整性机制是关系型数据库系统的核心功能。

关系完整性用于保证数据库中数据的正确性。系统在进行更新、插入或删除等操作时,都要检查关系完整性,以核实其约束条件,即关系模型的完整性规则。在关系模型中有四类完整性约束:实体完整性、参照完整性、域完整性和用户自定义完整性,其中实体完整性和参照完整性约束条件,称为“关系的两个不变性”。

1. 实体完整性

实体完整性指的是表中每一行的完整性。它主要用于确保操作的数据(记录)非空、且不重复(即实体完整性要求每个关系(表)有且仅有一个主键,每一个主键值不允许为“空”(NULL)或重复)。

在数据库中,空值表示值未知,且与空白或零值不同。没有两个相等的空值。比较两个空值或将空值与任何其他任何值比较时,结果通常是未知,因为空值的存在意味着数据的不确定性。

2. 参照完整性

1) 外键

设 F 是基本关系 R 中的一个或一组属性,但不是关系 R 的主键。如果 F 中的值在另一个基本关系 S 的主键 K 中有对应的值,则称 F 是基本关系 R 的外键。

例如,有如下两个表(学生表和成绩表):

学生表(学号, 姓名, 性别, 出生日期), 其中学号为主键
成绩表(成绩编号, 学号, 课程号, 成绩), 其中成绩编号为主键

在成绩表中,学号字段和课程号字段都是外键。外键与主键总是不可分的,其中主键所在的表称为主表,而外键所在的表则称为从表。

关系 R 和 S 可以表示不同的关系表,也可以表示同一个关系表,具体取决于它们之间的关系。以如下学生表为例:

学生表(学号, 姓名, 性别, 专业号, 年龄, 班长学号), 其中学号为主键

“班长学号”属性与学生表的“学号”属性相对应,因此“班长学号”是外键。在这种情况下,学生表既是主表又是外表。

说明:

外键不一定要与相应主键同名,但为了便于识别和理解,外键在设计时通常会与主键名称相同,特别是当它们属于不同的关系表时。

2) 参照完整性

在关系模型中,实体及实体间的联系都是通过关系来描述的,并且关系之间可能存在引用。参照的完整性要求关系中不能引用不存在的实体。参照完整性也称为引用完整性。

参照完整性是对相关联的两张表间的一种约束(也可以是对同一表内字段的引用)。用于确保不同表之间的数据一致性,避免因一张表数据的修改,导致另一张表相关数据失效。它通过检查主键和外键的取值,要求所有外键的值必须是主键的有效值,即外键的值要么来自主键,要么是空值。

3) 外键约束

在对数据库进行修改时,有可能会破坏表之间的参照完整性。因此,为了维护数据库中数据的完整性,应该对数据库的修改加以限制。这些限制包括插入约束、删除约束和更新约束。

- 插入约束。当在外表中插入新记录时,必须保证其中的外键值在主表中存在。例如,在学生表和成绩表之间用学号建立关联,学生表是主表,成绩表是从表。当向成绩表中输入一条新记录时,系统需要检查新输入的学号是否在学生表中已存在。如果存在,则允许执行插入操作;否则拒绝插入。如果没有参照完整性约束,从表中可能会输入主表中不存在的学号记录,这在现实中是不符合逻辑的。例如,一个不存在的学生却有学号记录,这是不符合常理的。
- 删除约束。当从主表中删除一条记录时,必须考虑外表数据的完整性。一般数据库提供两种常见的约束。一种是“限制删除”,即如果系统检查记录的主键值在某个外表中存在,则不允许删除;另一种是“级联删除”,即如果系统检查发现记录的主键值在某个外表中存在,则在删除主表记录的同时,将外表中与该主键值相对应的记录全部删除。
- 更新约束。如果要修改主表中的主键值,必须考虑外表数据的完整性。一般数据库提供两种约束:一种是“限制更新”,即如果系统检查发现记录的主键值在某个外表中存在,则不允许更新主表中的该记录;另一种是“级联更新”,即如果系统检查发现记录的主键值在某个外表存在,则在更新主表中该记录时,同时更新外表中所有引用该主键值的外键值。

提示:

级联更新和级联删除谨慎使用,因为级联操作会改变或删除从表的数据。

3. 域完整性

域完整性限制了某些属性中出现的值,将属性限制在一个有限的集合中。对于超出正常值范围的数据,系统将报警,同时这些非法数据不能进入数据库中。域完整性又称列完整性。域完整性通过限制数据类型、检查约束、默认值和非空属性的定义来确保数据的正确性。例如,对于“性别”字段的取值只能是“男”或“女”,在职工的年龄不能大于65岁等,这些都是针对具体关系提出的完整性条件。

4. 用户自定义完整性

实体完整性和参照完整性是关系模型必须满足的基本完整性约束条件,被称为关系的两个不变性,由关系数据库自动支持(适用于任何关系数据库系统)。除此之外,不同的关系数据库系统根据其应用环境的不同,往往还需要一些特殊的约束条件。

用户自定义完整性不属于其他任何完整性类别的特定业务规则。所有完整性类别都支持用户自定义完整性，包括 CREATE TABLE 中所有的列级约束和表级约束、存储过程以及触发器。

用户自定义完整性是指针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。实现用户自定义完整性常用的约束类型包括非空约束、唯一约束、检查约束、主键约束和外键约束等。

目前，多数关系数据库系统都提供了比较完善的约束机制。只要用户在定义表的结构时注意实体完整性和域完整性，建立表间关系时进行参照完整性约束的设置，数据库管理系统会自动维护这些完整性约束，以保证数据的完整性和一致性。

在关系数据库设计中，需要遵循各类完整性约束。在实际应用中，需要注意完整性约束之间可能的冲突。例如，参照完整性约束可能会与用户自定义完整性约束产生冲突。此外，还需要考虑完整性约束对性能的影响，因为过多的完整性约束可能会导致性能下降。

为维护数据的完整性，DBMS 必须能够：

- 提供定义完整性约束条件的机制，通常通过 SQL 的 DDL 语句实现，并作为数据库模式的一部分存储在数据字典中。
- 提供完整性检查的方法，在执行 INSERT、UPDATE、DELETE 语句时或在事务提交时进行检查，确保操作后的数据库数据不违背完整性约束条件。
- 违约处理情况，若 DBMS 发现用户的操作违背了完整性约束条件，它可以采取相应的措施，如拒绝执行操作，或在外键的约束下执行级联更新或删除操作，以维护关系的完整性。

1.3 数据库发展方向

随着云计算、大数据、人工智能等技术的不断发展，数据库系统也在不断演化和改进。本节将从以下几个方面对数据库未来的发展进行介绍。

1. 云数据库

随着云计算的兴起，越来越多的企业开始采用云上部署应用程序的方式，以便更好地管理应用程序和数据。因此，云数据库在未来有望成为主流。目前，主要的云计算提供商，如 AWS、Microsoft Azure 和 Google Cloud，已经提供了多种类型的云数据库服务，例如关系数据库、NoSQL 数据库和图形数据库等。

云数据库的优点有以下几个。

- (1) 灵活性：用户可以根据需要轻松地增加或减少数据库规模。
- (2) 高可用性：云数据库通常具有高可用性，可保证全天候(24/7)运行。
- (3) 安全性：云数据库通常具有高级安全功能，例如加密、身份验证和审计等。
- (4) 成本效益：云数据库通常使用按需计费模式，可以大大降低成本。

2. 大数据

当前，生产和收集的数据量正以指数级别增长。这些海量的数据需要被有效地管理和分析。在未来，大数据将继续成为数据库的关键发展方向。大数据技术可以使企业更好地管理和分析

数据，从而提高生产力和决策效率。随着大数据技术的不断发展，越来越多的数据库系统将支持大数据处理和管理。

3. 人工智能

另一个重要的数据库发展方向是与人工智能技术的集成。人工智能技术可以帮助企业更好地识别模式，并从大量数据中提取价值。例如，机器学习和深度学习技术可以将数据用于预测、分类和聚类等任务。因此，在未来，数据库将需要更好地支持这些技术。一些数据库已经开始支持机器学习和深度学习算法，例如 Apache Spark 和 Apache Hadoop 等。

4. 区块链

区块链技术也将对数据库的未来发展产生影响。区块链可以帮助企业构建去中心化的数据库系统，实现更高的安全性和可靠性。例如，分布式账本技术可以确保数据的不可篡改性和完整性。因此，在未来可能会看到越来越多的数据库系统与区块链技术进行集成。

5. 自动化

数据库系统的自动化将成为未来发展的重要方向之一。随着企业数据量的不断增长，数据库管理已经变得越来越复杂。因此，自动化将成为未来数据库发展的一个趋势。未来的数据库系统可能将自动执行任务，例如优化查询、备份和恢复等。

综上所述，未来数据库系统的发展将朝着云数据库、大数据、人工智能、区块链和自动化等方向发展。这些技术的发展将有助于解决当前数据库系统面临的挑战，并提供更高效、安全和可靠的数据管理和分析服务。

1.4 非关系型数据库

非关系型数据库是近年来迅速发展的一种数据库技术。云计算、物联网等新一代技术的发展，在移动计算、社交网络等业务的推动下，大数据技术出现并迅速建立起生态体系。然而，大数据在推动技术变革的同时，企业对海量数据的存储和并发访问要求越来越高。传统关系数据库的 ACID 原则、结构规整以及表连接操作等特性，成为制约海量数据存储和并发访问的瓶颈。

非关系型数据库(NoSQL)是为了解决海量数据的存储、并发访问以及扩展而出现的，它具有数据模型灵活、并发访问度高、易于扩展和伸缩、开发效率高以及开发成本低等优点，能够应对大规模数据集合多重数据类型的挑战，尤其是大数据应用中，与传统的关系型数据库不同，非关系型数据库通常使用不同的数据模型和查询语言来存储和管理数据，如键值存储、文档存储、列族存储、图形存储等。

NoSQL 仅仅是一个概念，泛指非关系型的数据库。最常见的解释是“Non-Relational”，另一种解释是“Not Only SQL”，也被很多人接受。NoSQL 区别于关系数据库，它们不保证关系数据的 ACID 特性。

1.4.1 非关系型数据库的分类

目前对于非关系型数据库主要有四种数据存储类型：键值对存储(Key-Value)，列存储

(Column-Store), 文档存储(Document Store), 以及图形数据库(Graph Database)。每一种数据存储类型都可以解决相应的问题, 这些问题在某些情况下是关系型数据库难以高效解决的。

1. 键值对存储数据库

这一类数据库通常使用哈希表, 其中每个键对应一个指向特定数据的指针。键值对存储数据库的优势在于其简单性和易部署性。然而, 当数据库管理员(DBA)需要对部分值进行查询或更新时, 性能可能会受到限制。在这类数据库中, 如 Redis、Tokyo Cabinet/Tyrant、Voldemort 和 Oracle BDB, 都可能出现这种情况。

2. 基于列的数据库

基于列的数据库通常用于应对分布式存储的海量数据。传统的关系型数据库是基于行的, 每一行都带有一个行 ID 并且行中的每一个字段都存储在一张表中。基于列的数据库将每一列数据分开存储, 当查找一个数量较小的列的时候其查找速度较快, 键仍然存在, 但是它们的特点是指向了多个列。在基于列的数据库中, 增加一列新的数据是比较容易的, 因为现有的列不会受新增列影响。然而, 增加一整条记录可能需要更新多个列, 以确保数据的一致性。因此, 基于行的数据库在事务处理和实时更新方面通常优于基于列的数据库。基于列的数据库包括 Hbase、Cassandra 和 HyperTable。

3. 文档存储数据库

文档存储数据库适用于存储和管理文档, 其中文档通常是结构化的数据(如 JSON 格式)。文档存储数据库存储的文档可以是不同结构的, 常见的格式包括 JSON 和 BSON 等。文档存储数据库有 MongoDB 和 CouchDB 等。国内也有文档存储数据库, 如 SequoiaDB(已经开源)。

4. 图形数据库

图形数据库主要应用图形理论来存储实体间的关系信息, 并且能够扩展到多个服务器上。如, Neo4j、FlockDB、AllegroGrap 和 GraphDB。

NoSQL 数据库没有统一的查询语言(SQL), 因此进行数据库查询需要制定数据模型。许多 NoSQL 数据库提供 RESTful 数据接口或查询 API。

除上述数据库外, 为了打破国外技术封锁、掌握关键核心技术, 我国在自主研发数据库方面给予了政策支持。国产数据库百花齐放, 如 OceanBase、TBase、GaussDB 和 HYDATA。作为信息科技领域的核心组成部分, 这些数据库对于国家的信息安全、科技自主创新和经济的发展都起着重要作用。自主研发国产数据库可以促使国内企业在数据库领域进行技术创新, 不再完全依赖于国外技术, 这有助于提升国家的科技创新能力。知识学习是一场永无止境的旅程, 特别是在数据库领域, 每一个学习者都可以为国产数据库的研发做出贡献。因此, 积极学习数据库知识, 提高自身的专业素养, 不仅有助于个人的职业发展, 更能为国产数据库的研发做出实实在在的贡献。

1.4.2 非关系型数据库的比较

每一种非关系型数据库都有其独到之处。接下来, 通过表 1-1 对 NoSQL 数据库的 4 种类型进行比较。

表 1-1 NoSQL 数据库的比较

数据库类型	常见数据库	应用场景示例
键值对存储	Redis、Tokyo Cabinet/Tyrant、Voldemort、Oracle BDB	会话存储、网站购物车等
基于列	Hbase、Cassandra、Riak、HyperTable	日志记录、博客网站等
文档存储	MongoDB、CouchDB、RavenDB	内容管理应用程序、电子商务应用程序等
图形	Neo4j、FlockDB、AllegroGrap、GraphDB	欺诈检测、推荐应用等

1.5 本章小结

本章对数据库的入门知识进行了讲解。首先介绍了数据库的基础知识，包括数据库技术基础、计算机管理数据技术的发展、数据库系统的特点以及数据库管理系统的功能。

目前理论成熟且使用普及的模型是关系模型，本章详细介绍了关系运算和关系完整性，为后续的学习打下坚实的基础。

未来数据库系统的发展将朝着云数据库、大数据、人工智能、区块链和自动化等方向推进。这些技术的发展将有助于解决当前数据库系统面临的挑战，并提供更高效、安全和可靠的数据管理和分析服务。

非关系型数据库是近年来迅速发展的一种数据库技术。非关系型数据库(NoSQL)是为了解决海量数据的存储、并发访问和扩展等问题而出现的，它具有数据模型灵活、并发访问度高、易于扩展和伸缩、开发效率高以及开发成本低等优点，能够解决大规模数据集中多种数据类型的挑战，尤其是大数据应用难题。本章简要介绍了非关系型数据库的分类和比较，引导读者主动关注数据库技术发展的前沿动态。

1.6 本章习题

一、单选题

1. 用二维表来表示实体及实体之间联系的数据模型是()。
A. 实体-联系模型 B. 层次模型 C. 网状模型 D. 关系模型
2. 关系型数据库管理系统中所谓的关系是指()。
A. 各条记录中的数据彼此有一定的关系
B. 一个数据库文件与另一个数据库文件之间有一定的关系
C. 数据模型符合一定条件的二维表格式
D. 数据库中各个字段之间彼此有一定的关系

3. 使用 MySQL 按用户的应用需求设计的结构合理、使用方便、高效的数据库和配套的应用程序系统,属于一种()。
- A. 数据库
B. 数据库管理系统
C. 数据库应用系统
D. 数据模型
4. 数据库是()。
- A. 以一定的组织结构保存在计算机存储设备中的数据的集合
B. 一些数据的集合
C. 辅助存储器上的一个文件
D. 磁盘上的一个数据文件
5. 下列说法错误的是()。
- A. 人工管理阶段程序之间存在大量重复数据,数据冗余大
B. 文件系统阶段程序和数据有一定的独立性,数据文件可以长期保存
C. 数据库阶段提高了数据的共享性,减少了数据冗余
D. 上述说法都是错误的
6. MySQL 是一种关系型数据库管理系统,所谓的关系是指()。
- A. 一个数据库文件与另一个数据库文件之间有一定的关系
B. 数据模型符合一定条件的二维格式
C. 数据库中的实体存在的联系
D. 数据库中各实体的联系是唯一的
7. 下列描述中正确的是()。
- A. 数据库系统是一个独立的系统,不需要操作系统的支持
B. 数据库设计是指设计数据库管理系统
C. 数据库技术的根本目标是要解决数据共享的问题
D. 数据库系统中,数据的物理结构必须与逻辑结构一致
8. 在数据库中可以添加、编辑和删除表记录,这是因为数据库管理系统提供了()。
- A. 数据定义功能
B. 数据操作功能
C. 数据维护功能
D. 数据控制功能
9. 数据库系统的构成为:数据库、计算机系统、用户和()。
- A. 操作系统
B. 文件系统
C. 数据集合
D. 数据库管理系统
10. 关系数据库管理系统的3种基本关系运算不包括()。
- A. 比较
B. 选择
C. 连接
D. 投影
11. 数据库(DB)、数据库系统(DBS)和数据库管理系统(DBMS)之间的关系是()。
- A. DBMS 包括 DB 和 DBS
B. DBS 包括 DB 和 DBMS
C. DB 包括 DBS 和 DBMS
D. DB、DBS 和 DBMS 是平等关系
12. 在关系理论中,把二维表表头中的栏目称为()。
- A. 数据项
B. 元组
C. 结构名
D. 属性名

13. 下面有关关系数据库主要特点的叙述中, 错误的是()。
- 关系中每个属性必须是不可分割的数据单元
 - 关系中每一列元素必须是类型相同的元素
 - 同一关系中不能有相同的字段, 也不能有相同的记录
 - 关系的行、列次序不能任意交换, 否则会影响其信息内容
14. 以一定的组织方式存储在计算机存储设备上, 能为多个用户所共享的与应用程序彼此独立的相关数据的集合称为()。
- 数据库
 - 数据库系统
 - 数据库管理系统
 - 数据结构
15. 设有部门和职员两个实体, 每个职员只能属于一个部门, 一个部门可以有多名职员, 则部门与职员实体之间的联系类型是()。
- m:n
 - 1:m
 - m:k
 - 1:1
16. 在关系模型中, 实现“关系中不允许出现相同的元组”的约束是通过()。
- 候选键
 - 主键
 - 外键
 - 超键

二、填空题

- 关系数据库的任何查询操作都是由 3 种基本运算组合而成的, 它们是_____、_____和_____。
- 常见的数据模型有 3 种, 它们是_____、_____和_____。
- 数据库系统的核心是_____。
- 在数据库中能够唯一地标识一个元组的属性或属性的组合称为_____。
- 在关系数据的基本操作中, 把两个关系中相同属性值的元组连接到一起形成新的二维表的操作称为_____。
- 关系代数是一种关系操作语言, 它的操作对象和操作结果均为_____。
- 设关系 R 和 S 分别为 5 目和 4 目关系, 关系 T 是 R 和 S 广义的笛卡儿积, 则 T 的属性个数是_____, 记录个数是_____。
- 在教师表中, 如果要找出职称是“教授”的教师, 应该采用的关系运算是_____。
- 数据模型不仅反映的是事物本身的数据, 而且还表示事物之间的_____。
- 设有选修日语的学生关系 R , 选修德语的学生关系 S , 求选修了日语而没有选修德语的学生, 则需要进行的运算是_____。

三、实践题

设教学数据库中有如下三个关系, 请写出关系代数表达式。

- 学生关系: $S(SNO, SN, AGE, SEX)$
 - 课程关系: $C(CNO, CN, TEACHER)$
 - 学习关系: $SC(SNO, CNO, SCORE)$
- 查询课程号为 C3 的学生学号和成绩。
 - 查询课程号为 C4 的学生学号和姓名。
 - 查询学习了 MATHS 课程的学生的学号和姓名。

4. 查询学习课程号为 C1 或 C3 的课程的学生学号。
5. 查询不学习课程号为 C2 的课程的学生姓名和年龄。

四、思考题

1. 试述数据、数据库、数据库系统、数据库管理系统的概念。
2. 使用数据库系统有什么好处？
3. 试述数据库系统的特点。
4. 数据库管理系统的主要功能有哪些？